Text Sentiment Analysis

Assignment 2: *n-gram + classifier approach*

After reading and understanding the code, I ran it to get a first result, which I will try to improve. It was a F1-Score of 96, which is already not bad. First of all I printed the metrics of grid search to get a better understanding about which hyperparameter were chosen for the model. After that I added some paramters like max_features for CountVectorizer or adding (1,1) to n-gram range. I can't add unlimited parameters, because then GridSearch will take too long to find out the best hyperparameters. After doing that, I found out that (1,1) for n-gram so taking every word alone work better for the given data. After that I also tried to use different Classifiers to get a better result. I tried MultinomialNB, which implements the naive Bayes algorithm for multinomially distributed data. That classifier worked worse and gave me a result of 93. I also tried the classifier SGDClassifier, but it worked worse as well and also gave an F1-Score of 93. So I changed back to SVC. After submiting a got a score of 0.92948 on kaggle

Assignment 3: *neural network based approach*

I started again with reading the code and understanding every line and writing comments. Then I did some preprocessing by removing stop words and unnecessary characters and lowercase everything. After that I reduced the vocabulary size, as we don't have that much data. This gave me already a little improvement but nothing significantly. Then I started playing with the hyperparameters. I was trying different values for keras.layers.Embedding(…) and keras.layers.LSTM(…) but that also didn't give me a bigger improvement. Then I wanted to change the network to a conventional neural network. The difference between a RNN and a CNN is that an RNN is trained to recognize patterns across time, while a CNN learns to recognize patterns across space. Keras offers again various Convolutional layers, chose the Conv1D layer, because it ist he best for our task. This layer has again various parameters to choose from. After applying that I got a score of 88, which ist an improvement of 2% to the 86 at the beginning. I was also trying to chose the best hyperparameters, but couldn't get a significant improvement. After submiting a got a score of 0.87396 on kaggle

Sources:

https://deepai.org/machine-learning-glossary-and-terms/gradient-clipping

https://keras.io/optimizers/

https://www.tensorflow.org/api_docs/python

https://docs.scipy.org/doc/numpy/reference/generated/numpy.ndarray.shape.html

https://www.kaggle.com/ngyptr/lstm-sentiment-analysis-keras

https://realpython.com/python-keras-text-classification/

Questions

For script A2, what is the purpose of `class_weight='balanced'` in cell #3?

➔ The “balanced” mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data

In script A2, `CountVectorizer` is responsible for preprocessing. How can we add stop word removal by using this class? How can we change word-based n-gram to char-based ngram using this class?

➔ We can use the paramter „stop_words to“ do that. Its default value is „none“. We can write „english“ to use a build-in stop word list or we can make our own list. We can change word-based it to char-based n-gram by setting the analyizer to „char“ instead of „word“.

In Assignment #1, we use `GridSearchCV` with cross-validation. In A2, are we still using cross-validation? If not, what did we use?

➔ We still use cross-validation „`cv=[(np.arange(0, split)`“

In script A3, what is the tensor shape of the output from 'keras.layers.CuDNNGRU(100)`?

➔ First of all this line isn’t compatible anymore with the current tensorflow version. We have to use „`model.add(keras.layers.LSTM(100))`“ instead.

➔ We can get this information with the command model.summary. It is „(None, 100)“

In script A3, what is the purpose of `clipnorm=4.` for the Adam optimizer? Why we need to do it?

➔ We use it to control gradient clipping. All parameter gradients will be clipped to a maximum norm of 4. This prevents any gradient to have norm greater than the threshold and thus the gradients are clipped. There is an introduced bias in the resulting values from the gradient, but gradient clipping can keep things stable.